

Minimum Mean Cycle Problem in Bidirected and Skew-Symmetric Graphs

Maxim A. Babenko * Alexander V. Karzanov †

February 2, 2008

Abstract

The problem of finding, in an edge-weighted bidirected graph $G = (V, E)$, a cycle with minimum mean weight of its edges generalizes similar problems for both directed and undirected graphs. (The problem is considered in two variants: for the cycles without repeated edges and for the cycles without repeated nodes.) In this note we develop an algorithm to solve this problem in $O(V^2 \min(V^2, E \log V))$ -time (to compare: the complexity of an improved version of Barahona's algorithm for undirected cycles is $O(V^4)$). Our algorithm is based on a certain general approach to minimum mean problems and uses, as a subroutine, Gabow's algorithm for the minimum weight 2-factor problem in a graph. The problem admits a reformulation in terms of regular cycles in a skew-symmetric graph.

Keywords: bidirected graph, skew-symmetric graph, minimum mean cycle.

AMS Subject Classification: 05C38, 05C85, 90C27.

1 Introduction

Among a variety of discrete optimization problems, an interesting class is formed by the problems consisting in finding an object (subset) of a given type in which the mean weight of an element is minimized. One of the most popular problems of this sort is the minimum mean cycle problem in an edge-weighted directed graph. The classical algorithm due to Karp [6], based on a dynamic programming approach, finds such a cycle in $O(nm)$ time. (Here and later on n and m denote the numbers of nodes and edges, respectively, in the input graph.) An analogous problem for cycles of an undirected graph can also be solved efficiently: Barahona [1] reduces it to a series of minimum weight \emptyset -join computations (see also [8, Sec 29.11]), and an improved version of his algorithm requires only $O(n)$ such computations.

*Dept. of Mechanics and Mathematics, Moscow State University, Vorob'yovy Gory, 119899 Moscow, Russia, *email:* mab@shade.msu.ru. Supported by RFBR grants 03-01-00475, 05-01-02803, and 06-01-00122.

†Institute for System Analysis, 9, Prospect 60 Let Oktyabrya, 117312 Moscow, Russia, *email:* sasha@cs.isa.ru. Supported by NWO-RFBR grant 047.011.2004.017 and RFBR grant 05-01-02805 CNRSL_a.

This note presents an algorithm with complexity $O(n^2 \min(n^2, m \log n))$ for a more general problem, namely, for finding a minimum mean cycle in an edge-weighted bidirected graph. (The concept of bidirected graph was introduced by Edmonds and Johnson [2]; more about bidirected graphs can be found in, e.g., [8].)

Recall that in a *bidirected* graph $G = (V, E)$ edges of three types are allowed: a usual directed edge, or an *arc*, that leaves one node and enters another one; an edge directed from *both* of its ends; and an edge directed to *both* of its ends. When both ends of an edge coincide, the edge becomes a loop; to slightly simplify our description we admit no loop entering and leaving its end node simultaneously (this will lead to no loss of generality in what follows).

A *walk* in G is an alternating sequence $P = (s = v_0, e_1, v_1, \dots, e_k, v_k = t)$ of nodes and edges such that each edge e_i connects nodes v_{i-1} and v_i , and for $i = 1, \dots, k-1$, the edges e_i, e_{i+1} form a *transit pair* at v_i , which means that one of e_i, e_{i+1} enters and the other leaves v_i . (Note that e_1 may enter s and e_k may leave t .) If, in addition, $k \geq 1$, $v_0 = v_k$ and the pair e_1, e_k is transit at v_0 , P is called a *cycle*. A cycle is called *node-simple* if all its nodes (except $v_0 = v_k$) are different, and *edge-simple* if all its edges are different. Note that G may contain edge-simple cycles but no node-simple ones.

Our main problem is stated as follows:

- (P) *Given a bidirected graph $G = (V, E)$ and a function $w : E \rightarrow \mathbb{R}$ of weights of edges, Find an edge-simple cycle C of G that minimizes the value $\bar{w}(C) := w(C)/|C|$.*

Here $w(C)$ denotes the total weight of edges, and $|C|$ the number of edges (the *length*) of C . So we deal with the *minimum mean edge-simple cycle problem* in an edge-weighted bidirected graph. Instead, one can state, in a similar way, the minimum mean *node-simple* problem in (G, w) , which also may find reasonable applications. However, the latter is easily reduced to (P) (and therefore, can be excluded from consideration in what follows). Indeed, modify the graph as follows: replace each node v by a pair of nodes v_1, v_2 , connecting them by an edge with zero weight from v_1 to v_2 , and each edge of the original graph that enters (leaves) v make entering v_1 (resp. leaving v_2). The obtained graph has $2n$ nodes and $m+n$ edges, all edge-simple cycles C' in it are node-simple and just correspond to the node-simple cycles C of the original graph; moreover, for corresponding C' and C , one has $w(C') = w(C)$ and $|C'| = 2|C|$.

In fact, the minimum mean cycle problem for directed graphs G becomes equivalent to (P) (as well as to its node-simple version) when G is considered as a bidirected graph. As for the undirected case, an undirected graph G can be turned into a bidirected one by assigning the orientation of each edge from both of its ends and by adding, for each node v , a loop with zero weight that enters v (twice). This reduces the undirected minimum mean cycle problem to problem (P) (but not to its node-simple version).

We show the following.

Theorem 1.1 *Problem (P) can be solved in $O(n^2 \min(n^2, m \log n))$ time.*

Another class of nonstandard graphs is formed by so-called skew-symmetric graphs — directed graphs with involutions on the nodes and on the arcs that change the orientation of each arc. (This class of graphs, under the name of *antisymmetrical digraphs*, was introduced by Tutte [9].) There is a close relationship between these and bidirected graphs, and

problem (P) is, in fact, equivalent to the problem of finding a regular cycle having minimum mean weight in a skew-symmetric graph with symmetric weights of arcs (precise definitions will be given later). As a consequence, we obtain an $O(n^2 \min(n^2, m \log n))$ -algorithm for the latter problem.

A natural approach to the minimum mean problem for an abstract family \mathcal{F} of subsets of a set consists in reducing this problem to a series of (auxiliary) problems of finding a member of \mathcal{F} with minimum total weight when the weight function is shifted by a constant. The shifts are chosen in such a way that the cardinalities of intermediate subsets in this process monotonically decrease. Therefore, the process is finite, provided that the auxiliary problem is well-solvable. In a more general approach, \mathcal{F} is extended to a family \mathcal{D} by adding certain sets represented as the disjoint union of members of \mathcal{F} . A clever choice of \mathcal{D} may simplify the auxiliary problem significantly, yielding an efficient algorithm for the original problem. Just this idea is applied in [1] where undirected cycles are extended to \emptyset -joins. (One more approach to a certain class of minimum mean problems and related topics are discussed in [7].)

In our case, the role of extended family \mathcal{D} plays the family of circulations that traverse every node of a given bidirected graph at most twice. Then the auxiliary problem becomes equivalent to the minimum weight 2-factor problem in a certain associated undirected graph. The latter can be solved in $O(n \min(n^2, m \log n))$ time by use of Gabow's algorithm [3]. This yields an algorithm with the desired complexity, taking into account that the number of iterations in the process is $O(n)$.

The above-mentioned general approach is described in Section 2 and a proof of Theorem 1.1 is given in Section 3. The related problem for skew-symmetric graphs is discussed in Section 4.

2 A General Approach

In a general setting, the minimum mean problem is formulated as follows:

- (M) *Given a family \mathcal{F} of nonempty subsets of a finite set E and a function $w : E \rightarrow \mathbb{R}$,
Find $X \in \mathcal{F}$ minimizing $\bar{w}(X) := w(X)/|X|$,*

where $w(X)$ denotes $\sum_{e \in X} w(e)$. For convenience the elements of \mathcal{F} are called *feasible sets*; so the goal is to find a feasible set with minimum mean weight.

Let us call a family \mathcal{D} of nonempty subsets of E a *disjoint extension* of \mathcal{F} if $\mathcal{F} \subseteq \mathcal{D}$ and each member X of \mathcal{D} can be represented as the disjoint union of feasible sets, i.e.,

$$(1) \quad X = X_1 \sqcup \dots \sqcup X_k \quad \text{for pairwise disjoint sets } X_1, \dots, X_k \in \mathcal{F}.$$

(To emphasize that the sets involved in the union are pairwise disjoint, we use notation \sqcup rather than \cup .) Let $s(\mathcal{D})$ denote the maximum cardinality (size) of a member of \mathcal{D} .

For an arbitrary disjoint extension \mathcal{D} of \mathcal{F} , problem (M) can be reduced to at most $s(\mathcal{D})$ problems of finding a set $X \in \mathcal{D}$ having minimum weight $w'(X)$ with respect to some other weight function w' , followed by one problem of constructing a decomposition (1) for the final set X . The idea is rather transparent and has been encountered in special cases (cf., e.g., [7, 1]).

Functions w' occurring in the desired reduction are obtained by shifting w by a constant. More precisely, we say that w' is the *shift* of w by $a \in \mathbb{R}$ if $w'(e) := w(e) - a$ for all $e \in E$. We also denote w' by w^a and call a the *shift number*. The method below relies on two easy properties:

- (2) $\overline{w}^a(X) = \overline{w}(X) - a$ for any nonempty subset $X \subseteq E$;
- (3) if $X = X_1 \sqcup \dots \sqcup X_k$ for nonempty subsets X_i , then $\overline{w}(X) \geq \min_i \overline{w}(X_i)$.

Method. In the beginning put $a := \max(w(e) : e \in E)$. Then iteratively construct a set X and update a as follows. At each iteration, take as X a member of \mathcal{D} with $w^a(X)$ minimum, by solving the corresponding minimum weight problem. Put $b := \overline{w}^a(X)$. If (i) $b = 0$, then the process terminates with constructing a decomposition (1) of X and outputting any of its members. Otherwise (ii) update $a := a + b$ and proceed with the next iteration.

Lemma 2.1 *The above method terminates in at most $s(\mathcal{D}) + 1$ iterations and outputs an optimal solution to problem (M).*

Proof.

First of all we observe that at each iteration there exists a set $Y \in \mathcal{D}$ satisfying $w^a(Y) \leq 0$. Indeed, the initial choice of a guarantees this property to hold at the first iteration. And for each iteration, we have $w^{a'}(X) = w^a(X) - b|X| = 0$, where a is the shift number at the beginning of this iteration, X is the set in \mathcal{D} found on it, $b = \overline{w}^a(X)$, and a' is the next shift number $a + b$. Then the property holds for the next iteration.

Thus, each number b in the process is nonpositive. Suppose $b = 0$ happens at some iteration. Then the set X found on this iteration satisfies $w^a(X) = 0$, and by the minimality of X , we have $w^a(Y) \geq 0$ for all $Y \in \mathcal{D}$; equivalently: $0 = \overline{w}^a(X) \leq \overline{w}^a(Y)$. Now (2) and (3) imply that $\overline{w}(X_i) = a$ for each member X_i in a decomposition of X , and that $\overline{w}(Y) \geq a$ for any $Y \in \mathcal{F}$ (taking into account that $\mathcal{F} \subseteq \mathcal{D}$). So X_i is an optimal solution to (M).

Finally, consider two consecutive iterations. Let a and X denote the shift numbers at the beginning of the former iteration and the set found on it, and let a' and X' denote similar objects on the latter one. Suppose that the latter iteration was not the last in the process; then $\overline{w}^{a'}(X') < 0$. Also $a' = a + \overline{w}^a(X)$ implies $\overline{w}^{a'}(X') = \overline{w}^a(X') - \overline{w}^a(X)$. Therefore,

$$w^a(X')/|X'| < w^a(X)/|X|.$$

This inequality is possible only if $|X'| < |X|$, in view of $w^a(X) \leq w^a(X')$ and $w^a(X) < 0$. Thus, the process is finite and the number of iterations does not exceed $s(\mathcal{D}) + 1$. \square

Remark. It is seen from the above proof that the number of iterations in the method is estimated via the minimum cardinality of a set $X \in \mathcal{D}$ minimizing $w^a(X)$ on the first iteration. Also the method can start with any initial shift number a for which one guarantees the existence of a set $Y \in \mathcal{D}$ with $w^a(Y) \leq 0$ (provided that such an a can be computed efficiently).

To illustrate the method, consider problem (M) for the family \mathcal{F} of simple cycles, or circuits, in a undirected graph $G = (V, E)$ with edge weights w (regarding circuits as edge

sets). Since the minimum weight problem for circuits is NP-hard when negative weights are possible, and therefore, $\mathcal{D} := \mathcal{F}$ is a bad choice, one has to take as \mathcal{D} some nontrivial disjoint extension of \mathcal{F} . Following [1], we put \mathcal{D} to be the family of nonempty \emptyset -joins in G . (Recall that for $T \subseteq V$ with $|T|$ even, a set $J \subseteq E$ is called a T -join if the set of nodes with odd degrees in (V, J) is exactly T .) Clearly a \emptyset -join is decomposed into pairwise (edge-) disjoint circuits; such a decomposition is constructed in linear time. Each iteration of the method in our case consists in finding a minimum weight \emptyset -join, which can be carried out in $O(n^3)$ time (see [8]). Also the number of iterations is $O(m)$. Hence the obtained algorithm runs in $O(n^3m)$ time.

In an improved version of this algorithm, the number of iterations reduces to $O(n)$, yielding the overall time bound $O(n^4)$. This relies on the existence (and the possibility to efficiently construct) an initial shift number a such that there exists a nonempty \emptyset -join J whose weight $w^a(J)$ is minimum and whose cardinality is $O(n)$, and moreover, this weight is nonpositive; cf. Remark above.

To compute the desired a , we order the edges of G by nondecreasing their weights:

$$w(e_1) \leq w(e_2) \leq \dots \leq w(e_m).$$

We find the minimum number k such that the graph $G_k = (V, \{e_1, \dots, e_k\})$ contains a circuit C . Put $a := \overline{w}(C)$; then $w^a(C) = 0$. Define

$$\begin{aligned} E^+ &:= \{e \in E \mid w^a(e) \geq 0\} \\ E^- &:= \{e \in E \mid w^a(e) < 0\} \end{aligned}$$

Let J be a nonempty \emptyset -join with $|J|$ minimum among all \emptyset -joins J' minimizing $w^a(J')$. The circuit C is also a \emptyset -join, hence $w^a(J) \leq w^a(C) = 0$. We assert that $|J| < 2n$. Indeed, since G_{k-1} is acyclic and, obviously, contains E^- , we have $|J \cap E^-| \leq |E^-| < k \leq n$. Also $|J \cap E^+| \leq n$. For otherwise the subgraph $(V, J \cap E^+)$ would contain a circuit C' . Then $J' := J - C'$ is a nonempty \emptyset -join satisfying $w^a(J') \leq w^a(J)$ and $|J'| < |J|$, contradicting the minimality of J . Thus, $|J| < 2n$, and a is as required.

3 Minimum Mean Cycles in Bidirected Graphs

In this section we specialize the method described in Section 2 to find a minimum mean edge-simple cycle in a bidirected graph $G = (V, E)$ with a weighting $w: E \rightarrow \mathbb{R}$ on the edges. For brevity we omit the adjective “edge-simple” in what follows.

Let us call a subset $X \subseteq E$ *balanced* if for each node v , the numbers of edges in X entering v and leaving v are equal (counting twice each doubly entering or doubly leaving loop at v , if any). In particular, the edge-set of any cycle is balanced. If each node is entered by at most two edges in X , we say that this balanced set is *small*. In a similar fashion, a cycle is called *small* if it passes each node at most twice, i.e., its edge-set is a small balanced set. Balanced sets and small cycles are related by the following property (in fact, known in literature), which is a bidirected analog of the fact that a circulation in a digraph is decomposed into simple cycles.

Lemma 3.1 *Each balanced set X is representable as the union of pairwise edge-disjoint small cycles (regarding a cycle as an edge set).*

Proof.

Assuming $X \neq \emptyset$, choose a maximal edge-simple walk $P = (v_0, e_1, v_1, \dots, e_k, v_k)$ in the subgraph of G induced by X . The fact that X is balanced easily implies that $v_0 = v_k$ and that e_1, e_k form a transit pair at v_0 , i.e., P is a cycle.

Deleting the edges of P from X , we again obtain a balanced set. So it follows by induction that X is representable as the union of pairwise edge-disjoint cycles. Let the number of cycles in such a representation be as large as possible. We assert that each cycle is small.

Indeed, suppose some cycle $C = (v_0, e_1, v_1, \dots, e_k, v_k)$ among these is not small, i.e., C passes some node v at least three times. Since a cycle can be considered up to reversing and cyclically shifting, we may assume that $v = v_0 = v_i = v_j$ for some $0 < i < j < k$, and that e_1 leaves v_0 . If for some $p \in \{i, j\}$, the edge e_p enters v_p , then the pair e_1, e_p (as well as e_k, e_{p+1}) is transit at v , and we can split C into two cycles: the part of C from v_0 to v_p and the rest. And if e_i leaves v_i and e_j leaves v_j , then the pair e_{i+1}, e_j is transit, and C can again be split into two cycles. A contradiction. \square

It is straightforward to devise a linear time algorithm that decomposes a balanced set into small cycles.

An immediate consequence of (3) and Lemma 3.1 is that for any cycle C , there exists a small cycle C' such that $\bar{w}(C') \leq \bar{w}(C)$. Therefore, instead of all cycles, we can take as the family \mathcal{F} in problem (M) the collection of (the edge-sets of) small cycles. Also the desired disjoint extension \mathcal{D} of \mathcal{F} can be defined to be the collection of *small* balanced sets. The cardinality of any small balanced set X does not exceed $2n$ since each node is incident with at most four edges in X . This implies that the number of iterations of the method applied to these \mathcal{F} and \mathcal{D} is $O(n)$, by Lemma 2.1. Each iteration consists in finding a small balanced set X with minimum weight $w'(X)$ for a current weight function $w' : E \rightarrow \mathbb{R}$, and it remains to explain how to solve this problem.

We reduce it to the *minimum weight 2-factor problem* in a certain undirected multigraph $\tilde{G} = (\tilde{V}, \tilde{E})$, with possible loops, formed from G as follows. Each node $v \in V$ generates two nodes \tilde{v}_1, \tilde{v}_2 in \tilde{G} . Each edge e in E connecting nodes u and v generates an edge \tilde{e} with the same weight connecting nodes \tilde{u}_i and \tilde{v}_j , by the following rule: if e enters u then $i = 1$, otherwise $i = 2$, and similarly for v and j . In particular, a doubly entering loop at v (if any) induces an undirected loop at \tilde{v}_1 . Finally, for each $v \in V$, we add *two* parallel edges connecting \tilde{v}_1 and \tilde{v}_2 in \tilde{G} and assign them zero weight; these edges are called *auxiliary*. An example of this transformation is depicted in Fig. 1.

Recall that a *2-factor* in a (multi)graph is a subset of edges such that each node of the graph is covered by exactly two edges of this subset (counting a loop twice). There is a natural correspondence ϕ between the small balanced sets in G and the 2-factors in \tilde{G} (up to swapping parallel auxiliary edges), and this correspondence preserves set weights. More precisely, for a small balanced set X in G , $\phi(X)$ contains all edges of \tilde{G} generated by X . In addition, for each node $v \in V$ incident with d entering edges in X (counting a loop twice), $\phi(X)$ contains $2 - d$ auxiliary edges connecting \tilde{v}_1 and \tilde{v}_2 . One can check that $\phi(X)$ is indeed a 2-factor. Moreover, any 2-factor of \tilde{G} can be obtained in this way.

Applying the $O(n \min(n^2, m \log n))$ -algorithm due to Gabow [3] to find a minimum weight 2-factor in \tilde{G} , we conclude that each iteration in our method can be performed

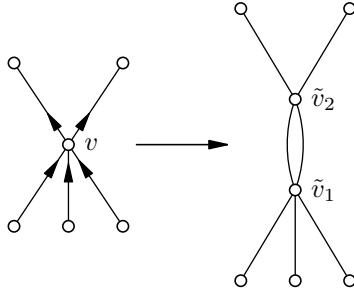


Figure 1: Reduction to 2-factor problem.

with a similar time bound. Since the number of iterations is $O(n)$, we obtain an algorithm to solve the original problem (P) in $O(n^2 \min(n^2, m \log n))$, as required in Theorem 1.1.

4 Skew-Symmetric Graphs

The above result on bidirected graphs can be translated into the language of skew-symmetric graphs. (About skew-symmetric graphs and problems on them, see, e.g., [9, 4, 5].)

A *skew-symmetric graph* is a digraph $G = (V, E)$, with possible multiple arcs, endowed with two bijections σ_V, σ_E such that: σ_V is an involution on the nodes (i.e., $\sigma_V(v) \neq v$ and $\sigma_V(\sigma_V(v)) = v$ for each node v); σ_E is an involution on the arcs; and for each arc a from u to v , $\sigma_E(a)$ is an arc from $\sigma_V(v)$ to $\sigma_V(u)$. For brevity, the mappings σ_V, σ_E are combined into one mapping σ on $V \cup E$, which is called the *symmetry* (rather than skew-symmetry) of G . For a node (arc) x , its symmetric node (arc) $\sigma(x)$ is also called the *mate* of x , and we will use notation with primes for mates, denoting $\sigma(x)$ by x' .

Observe that if G contains an arc a from a node v to its mate v' , then a' is also an arc from v to v' .

The symmetry σ is extended in a natural way to walks, cycles and other objects in G . In particular, two walks or cycles are symmetric to each other if the elements of one of them are symmetric to those of the other and go in the reverse order: for a walk $P = (v_0, a_1, v_1, \dots, a_k, v_k)$, the symmetric walk $\sigma(P)$ is $(v'_k, a'_k, v'_{k-1}, \dots, a'_1, v'_0)$.

Next we explain a relationship between skew-symmetric and bidirected graphs (cf. [5, Sec. 2]). Given a skew-symmetric graph $G = (V, E)$, choose an arbitrary partition $\pi = \{V_1, V_2\}$ of V such that V_2 is symmetric to V_1 . Then G and π determine the bidirected graph $\overline{G} = (\overline{V}, \overline{E})$ with $\overline{V} := V_1$ whose edges correspond to the pairs of symmetric arcs in G . More precisely, arc mates a, a' of G generate one edge e of \overline{G} connecting nodes $u, v \in V_1$ such that: (i) e goes from u to v if one of a, a' goes from u to v (and the other goes from v' to u' in V_2); (ii) e leaves both u, v if one of a, a' goes from u to v' (and the other from v to u'); (iii) e enters both u, v if one of a, a' goes from u' to v (and the other from v' to u). In particular, e is a loop if a, a' connect a pair of symmetric nodes.

Conversely, a bidirected graph $\overline{G} = (\overline{V}, \overline{E})$ determines a skew-symmetric graph $G = (V, E)$ with symmetry σ as follows. Take a copy $\sigma(v)$ of each element v of \overline{V} , forming the set $\overline{V}' := \{\sigma(v) \mid v \in \overline{V}\}$. Now set $V := \overline{V} \sqcup \overline{V}'$. For each edge e of \overline{G} connecting nodes u and v , assign two “symmetric” arcs a, a' in G so as to satisfy (i)–(iii) above (where $u' = \sigma(u)$

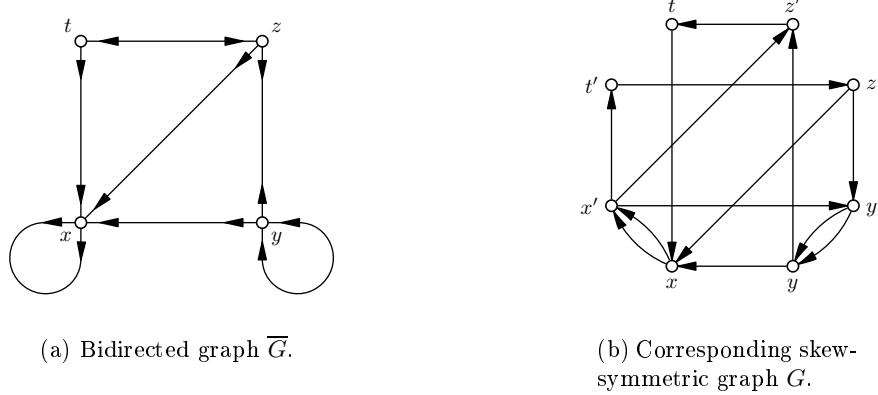


Figure 2: Related bidirected and skew-symmetric graphs.

and $v' = \sigma(v)$). An example is depicted in Fig. 2.

(Note that one bidirected graph generates one skew-symmetric graph, by the second construction. On the other hand, one skew-symmetric graph generates a set of bidirected graphs, depending on the partition π of V , by the first construction. One can check that among these bidirected graphs, one is obtained from another by the following operation: choose a subset X of nodes, and for each node v in X and each edge e incident with v , reverse the direction of e at v . This implies that the sets of walks (cycles) in these bidirected graphs are the same.)

There is a one-to-one correspondence between walks in \bar{G} and (directed) walks in G . More precisely, let τ be the natural mapping of $V \cup E$ to $\bar{V} \cup \bar{E}$ (obtained by identifying the pairs of symmetric nodes and arcs). Each walk $P = (v_0, a_1, v_1, \dots, a_k, v_k)$ in G (where a_i is an arc from v_{i-1} to v_i) induces the sequence

$$\tau(P) := (\tau(v_0), \tau(a_1), \tau(v_1), \dots, \tau(a_k), \tau(v_k))$$

of nodes and edges in \bar{G} . One can see that $\tau(P)$ is a walk in \bar{G} and that $\tau(P')$ is the walk reverse to $\tau(P)$. Moreover, for any walk \bar{P} in \bar{G} , there is exactly one preimage $\tau^{-1}(\bar{P})$.

Finally, following terminology of [4], a walk (cycle) in a skew-symmetric graph is called *regular* if it is arc-simple and contains no pair of symmetric arcs (while pairs of symmetric nodes in it are allowed). Certain problems on regular walks and cycles are studied in [9, 4] (and some other works). One more problem is:

- (S) *Given a skew-symmetric graph $G = (V, E)$ and a symmetric weight function $w : E \rightarrow \mathbb{R}$ (i.e., $w(a) = w(a')$ for all $a \in E$), find a regular cycle C in G whose mean weight $\bar{w}(C)$ is minimum.*

It is not difficult to check that for each regular cycle C in G , the corresponding cycle $\tau(C)$ in the bidirected graph \bar{G} is edge-simple, and vice versa. Moreover, the weights (and the mean weights) of the corresponding cycles are equal, assuming that for $e \in E$, the weight of $\tau(e)$ is defined to be $w(e)$ (this is well-defined since w is symmetric). Therefore, problem (S) is equivalent to (P), and we obtain the following.

Corollary 4.1 Problem (S) can be solved in $O(n^2 \min(n^2, m \log n))$ time.

References

- [1] F. Barahona, *Reducing matching to polynomial size linear programming*, SIAM Journal on Optimization **3** (1993), 688–695.
- [2] J. Edmonds and E. L. Johnson, *Matching, a well-solved class of integer linear programs*, Combinatorial Structures and Their Applications (1970), 89–92.
- [3] H. N. Gabow, *An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems*, STOC, vol. 15, 1983, pp. 448–456.
- [4] Andrew V. Goldberg and Alexander V. Karzanov, *Path problems in skew-symmetric graphs*, Combinatorica **16** (1996), no. 3, 353–382.
- [5] ———, *Maximum skew-symmetric flows and matchings*, Mathematical Programming **100** (2004), no. 3, 537–568.
- [6] R. M. Karp, *A characterization of the minimum cycle mean in a digraph*, Discrete Mathematics **23** (1978), 309–311.
- [7] A. V. Karzanov, *Minimum mean weight cuts and cycles in directed graphs*, Qualitative and Approximate Methods for Studying Operator Equations (1985), 72–83 (Russian).
- [8] A. Schrijver, *Combinatorial optimization*, vol. A, Springer, Berlin, 2003.
- [9] W. T. Tutte, *Antisymmetrical digraphs*, Canadian J. Math. **19** (1967), 1101–1117.